Выступление:  
**Тема**: Тестирующая система для школьников в мессенджере Telegram

Мой проект нацелен на упрощение учебного процесса, как для учеников, так и для учителей. Я хочу, чтобы учителя тратили меньше времени на проверку на составление, сбор, проверку работ школьников.

1. **Телеграм:**

Телеграм — это мессенджер (программа для обмена сообщениями), реализованный по клиент-серверной архитектуре. Используя сервер для создания диалога между двумя клиентами, Телеграм пересылает через него или напрямую текстовые сообщения, а также изображения, видео или документы других форматов.

1. **Телеграм-бот**

Представьте, что у нас есть зарегистрированный аккаунт. Все сообщения, присылаемые на него, попадают на вход некоторой программе, а выход этой программы отправляется в виде сообщения тому, от кого пришло исходное сообщение. Это и есть бот. Иными словами, Телеграм-бот — это специальный пользователь, поведением которого управляет некоторая программа.  
Примеры: Боты-ответчики (вики), интерфейс доступа к веб-сервисам, игры,

1. **API**

Давайте представим себе, что мы с вами разработали какую-нибудь очень полезную программу. Программой пользуется все большее количество людей. У нас, конечно же, есть канал обратной связи, по которому пользователи сообщают нам об ошибках в программе и каких функций им не хватает. Мы исправляем ошибки и стараемся расширять функциональность. Но силы наши конечны, а пожеланий с ростом популярности программы становится все больше и больше. Делать все мы не успеваем. Да и нет смысла выполнять каждое пожелание, если мы понимаем, что данная функция нужна лишь небольшой аудитории. Мы можем написать библиотеку, дающую доступ к некоторым функциям нашей программы. Понятно, что таким образом мы дадим инструмент только программистам, но программиста можно нанять, и это уже не обязательно должен быть наш сотрудник. Если мы, помимо библиотеки, напишем инструкцию к ней, опишем ее функции, как ими пользоваться и как построить приложение на основе нашей библиотеки, то сторонний программист вполне сможет разобраться с ней и решить задачу заказчика.

1. **VCS**

VCS **-** программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

1. Сохраняют код при поломках
2. Хранят много версий кода программы и позволяют легко переключаться между ними
3. Помогают разработчикам обмениваться кодом и редактировать один и тот же код с разных устройств
4. Объединяют результаты труда нескольких разработчиков

Чаще всего используют системы контроля версий Git, Mercurial (ее еще называют Hg) и несколько устаревшие Subversion (она же SVN) и CVS. У каждой из них есть свои плюсы и минусы, я же использовал систему **Git**. Ознакомившись с ней, я создал своё репозиторий и по мере написания кода сохранял в ней промежуточный результат.

1. **SQLite**
2. **Python**
3. **Библиотеки**

* requests
* PyTelegramBotAPI
* os
* threading
* time
* datetime
* PyQt5